③

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-051890
(43)Date of publication of application : 23.02.2001

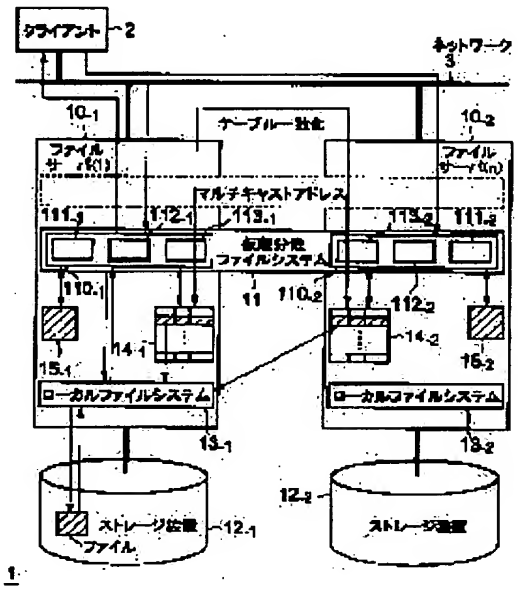
(51)Int.Cl. G06F 12/00
G06F 13/00

(21)Application number : 11-226494
(22)Date of filing : 10.08.1999

(71)Applicant : TOSHIBA CORP
(72)Inventor : UCHIBORI IKUO
TAKAKUWA MASAYUKI

(54) VIRTUAL DECENTRALIZED FILE SERVER SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To make a client not pay attention to the number of file servers decentralized in a network and the connection states of storage devices.
SOLUTION: This virtual decentralized file server system 1 is equipped with servers 10-1 and 10-2 decentralized in the network 3 and a virtual decentralized file system 11 is decentralized and mounted on each of the servers. Modules 110-1 and 110-2 on the servers 10-1 and 10-2 which constitute this system 11 when receiving a file operation request multicast from a client 2 judge whether or not their servers are optimum servers capable of handling the request according to server information holding parts 15-1 and 15-2 holding mapping tables 14-1 and 14-2 between the virtual decentralized file system 11 and all local file systems 13-1 and 13-2 or server information on all the servers, and makes a local file system of a corresponding server perform requested file operation according to the judgement result.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Japanese Patent Laid-open Publication No. 2001-51890 A

Publication date: February 23, 2001

Applicant:: TOSHIBA CORP

Title: Virtual distributed file server system

(57) [Abstract]

[Object] To make it unnecessary that clients consider the number of file servers distributed on a network and the connection state of storage devices.

[Means] Distributed servers 10-1 and 10-2 are provided on a network 3, and a virtual distributed file system 11 is distributed and mounted on each server. When modules 110-1 and 110-2 on the servers 10-1 and 10-2 constituting this system 11 receive a multi-cast file operation request from a client 2, determine whether the own server is the optimum server capable of processing the request, based on mapping tables 14-1, 14-2 between the virtual distributed file system 11 and all local file systems 13-1 and 13-2, or server information holding sections 15-1, 15-2 that hold server information of all servers, and allow a local file system of the corresponding server to perform the requested file operation based on the determination result.

[What is claimed is]

[Claim 1] A virtual distributed file server system comprising a plurality of file servers capable of multi-casting and distributed on a network, comprising:

a virtual distributed file system distributed and mounted on the respective file servers and independent of the actual storage configuration, which manages files in all file servers in an integrated manner;

a local file system mounted on each filer server respectively independently, for

managing the storage configuration peculiar to the respective servers; and

a mapping table provided in the respective file servers for holding mapping information between the virtual distributed file server system and the local file system that actually manages files, with respect to respective files managed by the virtual distributed file system in an integrated manner;

wherein the virtual distributed file system comprises management modules respectively provided in the respective file servers, and each management module commonly receives a multi-cast file operation request from a client, and refers to the mapping table in the own server in response to the request, to determine whether the own server is the optimum server capable of processing the request, and only when it determines that the own server is the optimum server, allows the local file system in the corresponding server to perform the requested file operation.

[Claim 2] A virtual distributed file server system comprising a plurality of file servers capable of multi-casting and distributed on a network, comprising:

a virtual distributed file system distributed and mounted on the respective file servers and independent of the actual storage configuration, which manages files in all file servers in an integrated manner;

a local file system mounted on each filer server respectively independently, for managing the storage configuration peculiar to the respective servers;

a mapping table provided in the respective file servers for holding mapping information between the virtual distributed file server system and the local file system that actually manages files, with respect to respective files managed by the virtual distributed file system in an integrated manner; and

a server information holding unit provided in the respective file servers, which holds, with respect to all file servers, server information including at least one of

information indicating unused capacity of a storage device in the server, and information indicating the load status of the server,

wherein the virtual distributed file system comprises management modules respectively provided in the respective file servers, and each management module commonly receives a multi-cast file operation request from a client, and refers to the mapping table in the own server or the server information holding unit in response to the request, to determine whether the own server is the optimum server capable of processing the request, and only when it determines that the own server is the optimum server, allows the local file system in the corresponding server to perform the requested file operation.

[Claim 3] The virtual distributed file server system according to claim 1 or 2, wherein when the file operation request is a file read request or a file write request, the management module refers to the mapping table in the own server, to determine whether the own server is the optimum server capable of processing the request, according to whether the corresponding file is under management of the local file system in the own server.

[Claim 4] The virtual distributed file server system according to claim 2, wherein when the file operation request is a new file creation request, the management module refers to the server information holding unit in the own server, to compare the unused capacity of the storage device of the server or the load status of the server, for each of all servers, thereby determining whether the own server is the optimum server capable of processing the request.

[Claim 5] The virtual distributed file server system according to claim 1, wherein the management modules exchange the information of the mapping table in the own server and the information of the mapping table in the other servers by inter-server

communication, in order to make the contents of the mapping tables in all file servers agree with each other.

[Claim 6] The virtual distributed file server system according to claim 2, wherein the management modules exchange the information of the mapping table in the own server and the information of the mapping table in the other servers by inter-server communication, in order to make the contents of the mapping tables in all file servers agree with each other, and exchange the information of the server information holding unit in the own server and the information of the server information holding unit in the other servers by inter-server communication, in order to make the contents of the server information holding unit in all file servers agree with each other.

[Claim 7] The virtual distributed file server system according to claim 1 or 2, further comprising an each file load status information holding unit respectively provided in the respective file servers for holding the information indicating the load status for each file under management of the server, wherein

the management module detects a file having a load exceeding a first threshold from information held by the each file load status information holding unit in the own server, to replicate the file with respect to another optional file server by inter-server communication, and when there is a multi-cast read request of the file from a client, entrusts the replicated side with the processing with respect to the request.

[0004]

[Problems to be solved by the Invention]

In the computer network system, when files are shared by using a file server, it is general that the client side can see the volume configuration on the file server side. For example, when the server side increases a disk drive, the client side must

recognize the new volume and mount the volume. When the server itself is increased, complicated operation such as determination of the operation policy of the increased server or system setup and management becomes necessary, and the client side must recognize the new server and mount the new volume.

[0005] In the file sharing system using the conventional file server (file server system), when an increase of a disk drive (storage device) or an increase of a server is required, there is a problem in that great cost is required for new setup and management both on the server and client sides. Further, according to the operation mode of the storage, it may be desired to expand only the capacity of a specific file system, or only an increase of the storage device and the server may not solve the problem.

[0006] The present invention has been achieved in order to solve the above problems. It is an object of the present invention to provide a virtual distributed file server system in which clients can use a plurality of file servers distributed on a network as a single server, and clients are unnecessary to consider the number of servers and the connection state of the storage devices.

[0028]

[First Embodiment]

Fig. 1 is a block diagram illustrating the configuration of a computer network system to which the virtual distributed file server system according to the first embodiment of the present invention is applied.

[0029] In this figure, reference sign 1 denotes a virtual distributed file server system, and 2 denotes a client (a client computer) that requests file service to the file server system 1. The virtual distributed file server system 1 is realized by using a plurality of, for example, two file servers (server computers) 10-1, 10-2 distributed on a network 3.

In this figure, for the convenience sake, only one client 2 is shown, but it is general that a plurality of clients exists.

[0030] Reference sign 11 denotes a virtual distributed file system, being the center of the virtual distributed file server system 1, which is distributed and mounted on the respective file servers 10-1 and 10-2. The virtual distributed file system 11 manages files in all file servers 10-1 and 10-2 in an integrated manner, and provides the client 2 with a virtual file system that does not depend on the actual volume configuration (storage configuration) of the respective file servers 10-1 and 10-2.

[0031] The virtual distributed file system 11 has virtual distributed file modules 110-1 and 110-2 distributed and mounted on the respective filer servers 10-1 and 10-2. The virtual distributed file modules 110-1 and 110-2 are management modules for virtually making the file servers look like one file system with respect to the client 2, while distributing and processing the request from the client 2 on the file servers 10-1 and 10-2. The virtual distributed file modules 110-1 and 110-2 have virtual distributed file interfaces 111-1 and 111-2, which is the center of the modules 110-1 and 110-2, and processes the request from the client 2, interfaces (local file interfaces) 112-1 and 112-2 between local file systems 13-1 and 13-2 and the modules, and communication modules 113-1 and 113-2 that performs communication between the modules 110-1 and 110-2 (communication between servers represented by communication for making the information of the mapping tables 14-1, 14-2 and the server information agree with each other).

[0032] The file servers 10-1 and 10-2 are connected to the client 2 via the network 3. The file servers 10-1 and 10-2 are mounted with local file systems 13-1 and 13-2 that manage the storage devices 12-1 and 12-2 (actual storage configuration), such as disk drives respectively connected to the servers 10-1 to 10-n, in addition to the virtual

distributed file system 11.

[0033] The file servers 10-1 and 10-2 are provided with the mapping tables 14-1 and 14-2 having the same content for associating the virtual distributed file server system 111 with the local file systems 13-1 and 13-2. The data structure of the table 14-i (i = 1, 2) is shown in Fig. 2.

[0034] Each entry to the table 14-i includes a file name registration field (file name field) 141 of a file managed by the virtual distributed file system 1, a registration field (a virtual path field) 142 of a path (a virtual path) expressing a logical position (which can be seen from the client 2) of the file on the virtual distributed file system 1, a registration field (position information field) 143 of position information expressing a physical position (which cannot be seen from the client 2) of the file on the storage, a registration field (permission information field) 144 of permission information for managing the access right (permission/prohibition) to the file, and a registration field 145 of other various attributes.

[0035] The (virtual distributed file module 110-i on the) virtual distributed file system 11 refers to the mapping table 14-i having such a data structure, thereby obtaining location information, for example, in which file server 10-1 or 10-2 a certain file exists, and also obtain the attribute of the file, such as permission, according to need.

[0036] The file servers 10-1 and 10-2 are provided with server information holding sections 15-1 and 15-2 having the same content. The server information holding section 15-i (i = 1, 2) is used, as shown in Fig. 3, for holding server information including information indicating the unused storage capacity (resource information) of (the storage devices 12-1, 12-2 of) all file servers 10-1 and 10-2 constituting the virtual distributed file server system 1, and information indicating the load status.

[0037] The operation of the configuration shown in Fig. 1 will be explained next. In

the embodiment, it seems to the client 2 that not the local file systems 13-1 and 13-2 of the respective file servers 10-1 and 10-2, but the virtual distributed file system 11 is mounted. Therefore, when any file operation request occurs, the client 2 issues the same request to all file servers 10-1 and 10-2 on which the virtual distributed file system 11 is mounted. In this case, for example, according to a method such as using Internet protocol (IP) multi-cast, the client 2 side can issue a request without being aware of the number of the file servers.

[0038] Upon reception of the request from the client 2, the file servers 10-1 and 10-2 hand over the request to virtual distributed file modules 110-1 and 110-2 corresponding to the own server in the virtual distributed file system 11. (The virtual distributed file interfaces 111-1 and 111-2 in) the modules 110-1 and 110-2 determine the request type, whether the request is a file read request or a file write (update) request, or a new file creation request or a directory creation request.

[0039] It is assumed herein that the file operation request from the client 2 is a file read request or a file write request. The request is added with a file name of the requested file, and a path (virtual path) of the file on the virtual distributed file system 11.

[0040] When the file operation request from the client 2 is a file read request or a file write request, (the virtual distributed file interfaces 111-1 and 111-2 in) the virtual distributed file modules 110-1 and 110-2 refer to the mapping tables 14-1, 14-2 in the own server, by the file name and the virtual path of the requested file, to check if the operation-requested file is held in the own server (in the storage devices 12-1, 12-2 connected to the own server), based on the file name and the registration information of the position information field 143 in the entry of the tables 14-1 and 14-2.

[0041] If the requested file is held in the own server, (the virtual distributed file interfaces 111-1, 111-2 in) the virtual distributed file modules 110-1, 110-2 access the

actual file via the local file systems 13-1, 13-2 in the own server by the local file interfaces 111-1, 111-2, and give a response to the client 2. On the other hand, if the operation-requested file is not held in the own server, the virtual distributed file modules 110-1, 110-2 regard that another server is to respond, and do not give a response.

[0042] On the other hand, if the request from the client 2 is a new file creation or a directory creation, the virtual distributed file modules 110-1, 110-2 refer to the server information holding sections 15-1, 15-2 (not to the mapping tables 14-1, 14-2). According to a predetermined algorithm, based on the server information of all servers held in the server information holding sections 15-1, 15-2, only the virtual distributed file modules 110-i on either one server 10-i (i is 1 or 2) accepts the request from the client 2. Specifically, the virtual distributed file modules 110-i on the server 10-i compares the unused storage capacities indicated by the server information of all servers, and when it can be determined that the unused storage capacity of (the storage device 12-i of) the own server 10-i is the largest, accepts the request from the client 2. In this case, it is not always necessary that the information of the unused storage capacity of the corresponding server is included in the server information.

[0043] The virtual distributed file module 110-i may compare the loads indicated by the server information of al servers, and when the load of the own server is the lowest, may accept the request from the client 2. In this case, it is not always necessary that the information of the unused storage capacity of the corresponding server is included in the server information.

[0044] Other than this configuration, the virtual distributed file module 110-i may determine continuous areas that can be secured on the respective storage devices 12-1, 12-2, from the mapping information for each file in the mapping table 14-i (that is, from the status of use of the area in the respective storage devices 12-1, 12-2), and

when continuous areas not smaller than the necessary size can be secured, and the storage device having the largest size is the storage device 12-i in the own server, may accept the request from the client 2. In this case, the server information holding sections 15-1, 15-2 may not be always necessary.

[0045] Further, an evaluation value may be obtained under the condition of at least two of the unused storage capacity, the load status, and the size of the securable continuous areas (complex condition), to determine whether the own server is the optimum server for accepting the request.

[0046] In the virtual distributed file module 110-i on the file server 10-i, upon reception of the request from the client 2 by the virtual distributed file interface 111-i, the local file system 13-i creates a requested new file or a directory via the local file interface 112-i, thereby registering the corresponding entry information in the mapping tables 14-1 and 14-2.

[0047] After the new file or the directory has been created, the virtual distributed file module 110-i on the file server 10-i transmits the new entry information registered in the mapping table 14-i on the own server to the virtual distributed file module 110-j (j is 1 or 2, provided that $j \neq i$) on all other servers 10-j, by the communication module 113-i via the network 3. (The virtual distributed file interface 111-j in) the virtual distributed file module 110-j receives the entry information for the mapping table 14-i transmitted from the virtual distributed file module 110-i, and registers the entry information for the mapping table 14-i of the received other server 10-i, in the mapping table 14-j in the own server. Thus, since the virtual distributed file modules 110-1 and 110-2 on the file servers 10-1 and 110-2 exchange the newly registered (and updated) entry information for the mapping tables 14-1 and 14-2 with each other, the contents in the mapping tables 14-1 and 14-2 can be made to agree with each other.

[0048] The virtual distributed file modules 110-1 and 110-2 on the file servers 10-1 and 10-2 regularly update the server information (unused storage capacity and load status) of the own server, of the server information of the own server held by the server information holding sections 15-1 and 15-2 on the own server, and regularly transmit the updated server information to (the virtual distributed file modules 110-1 and 110-2 on) all other servers, by the communication modules 113-1, 113-2) via the network 3, thereby making the contents of the server information holding sections 15-1 and 15-2 in the respective file servers 10-1 and 10-2 agree with each other. In other words, the virtual distributed file modules 110-1 and 110-2 make the contents agree with each other by exchanging the server information regularly.

[0049] By the above operation, the file servers 10-1 and 10-2 can be autonomously distributed and made to cooperate with each other. As a result, the client 2 can be provided with a virtual file server, without noticing that two (a plurality of) file servers actually exist.

[0050] In the system example in Fig. 1, a system having two servers has been explained, but even when the system includes three or more servers, the virtual file server can be also provided by the similar mechanism.

[0051]

[Second Embodiment]

Fig. 4 is a block diagram illustrating the configuration of a computer network system to which the virtual distributed file server system according to the second embodiment of the present invention is applied, wherein like reference signs refer to like parts in Fig. 1.

[0052] In Fig. 4, a virtual distributed file system 41, being the center of a virtual distributed file server system 4, is distributed and mounted on n file servers 10-1 to

10-n. As the virtual distributed file system 11 in Fig. 1, the virtual distributed file system 41 manages files in all file servers 10-1 to 10-n in an integrated manner, and provides the client 2 with a virtual file system that does not depend on the actual volume configuration of the respective file servers 10-1 to 10-n. The virtual distributed file system 41 has virtual distributed file modules 410-1 and 410-n that process the request from the client 2 on the respective file servers 10-1 to 10-n. The modules 410-1 and 410-n have the same configuration as the modules 110-1 and 110-2 in Fig. 1, and includes virtual distributed file interfaces 411-1 to 411-n, local file interfaces 412-1 to 412-n, and communication modules 413-1 to 413-n. However, the communication modules 413-1 to 413-n in this embodiment are different from the communication modules 113-1 and 113-2 in Fig. 1, and constructed so as to perform communication via a private communication path 5 described later.

[0053] The file servers 10-1 to 10-n are connected with the client 2 via the network 3. The file servers 10-1 to 10-n are mounted with local file systems (local file systems) 13-1 to 13-2 that manage the storage devices 12-1 to 12-2 respectively connected to the corresponding servers 10-1 to 10-n. The file servers 10-1 to 10-n are provided with the mapping tables 14-1 to 14-n and the server information holding sections 15-1 to 15-2.

[0054] The characteristic points of the virtual distributed file server system 4 having the configuration shown in Fig. 4 are that the number of file servers constituting the system is n, and that the n file servers 10-1 to 10-n are connected with each other by the private communication path 5 separately from the network 3, different from the virtual distributed file server system 1 shown in Fig. 1. The private communication path 5 is, for example, Ethernet or Fibre Channel, but the physical layer is not particularly specified. As for the topology, the bus type topology is assumed in the

example in Fig. 4, but it may be a loop or a switch.

[0055] In the configuration in Fig. 4, in order that the file servers 10-1 to 10-n perform distribution and cooperation operation (by the virtual distributed file modules 410-1 to 410-n in the virtual distributed file system 41), it is necessary to make the contents of the mapping tables 14-1 to 14-n and the server information holding sections 15-1 to 15-n agree with each other between respective servers 10-1 to 10-n at all times, as can be figured out from the explanation in the first embodiment. However, if the agreement of information between the servers 10-1 to 10-n is realized via the network 3 as in the first embodiment, when the number of file servers constituting the virtual distributed file server system 4 increases, the traffic (in the communication between servers) for making the information agree with each other increases, thereby deteriorating the throughput on the network 3.

[0056] Therefore, in the second embodiment, as in the configuration shown in Fig. 4, the private communication path 5 exclusive for information exchange between servers is provided between the respective file servers 10-1 to 10-n. The private communication path 5 is used for inter-server communication performed by the communication modules 413-1 to 413-n in the virtual distributed file modules 410-1 to 410-n in the virtual distributed file system 41, that is, for inter-server communication for making the contents of the mapping tables 14-1 to 14-n and the server information holding sections 15-1 to 15-n agree with each other.

[0057] In this embodiment, since the private communication path 5 is used, instead of the network 3, for the inter-server communication for making the contents of the mapping tables 14-1 to 14-n and the server information holding sections 15-1 to 15-n agree with each other, the load on the network 3 can be reduced.

[0058]

[Third Embodiment]

In the first and the second embodiments, a configuration example of the virtual distributed file server system that distributes a plurality of file servers so as to cooperate with each other is shown. The configuration shown in Figs. 1 and 4 referred to in the first and the second embodiments is a static example with a specific number of servers. However, it is desired that the number of servers can be changed.

[0059] Therefore, the third embodiment of the present invention in which the number of servers constituting the virtual distributed file server system can be changed will be explained, with reference to the accompanying drawings. Fig. 5 is a block diagram illustrating the configuration of a computer network system to which the virtual distributed file server system according to the third embodiment of the present invention is applied, wherein like reference signs refer to like parts in Fig. 4.

[0060] First, it is assumed that a new file server 10-(n+1) is added, as shown in Fig. 5(a), to the virtual distributed file server system 4 shown in Fig. 4, that is, the virtual distributed file server system 4 including n file servers 10-1 to 10-n.

[0061] In this case, a virtual distributed file module 410-(n+1) on the virtual distributed file system 41 distributed also on the added file server 10-(n+1) locks the update of the entry information in the mapping tables 14-1 to 14-n, and the server information (including the resource information and the load status of each server) in the server information holding sections 15-1 to 15-n, as shown by reference sign A1 in Fig. 5(a), with respect to the file servers 10-1 to 10-n already constituting the virtual distributed file server system 4, by inter-server communication (for example, via a private communication path (not shown)).

[0062] The module 410-(n+1) on the added server 10-(n+1) copies the whole information in the mapping table 14-1 and the server information holding section 15-1,

as shown by reference sign A2 in Fig. 5(b), to the mapping table 14-(n+1) and the server information holding section 15-(n+1) in the own server, from any one server of other file servers 10-1 to 10-n, for example, from the file server 10-1, by inter-server communication.

[0063] The module 410-(n+1) on the added server 10-(n+1) adds the server information indicating the resource and load status of the own server to the copied server information holding sections 15-(n+1), as shown by reference sign A3 in Fig. 5(c).

[0064] Thereafter, the module 410-(n+1) on the added server 10-(n+1) issues a request for making the server information agree with each other with respect to all other file servers 10-1 to 10-n, as shown by reference sign A4 in Fig. 5(d), by inter-server communication, and then releases the lock.

[0065] By the series of operation, a new server (file server 10-(n+1)) can be added dynamically with respect to the virtual distributed file server system 4 already constructed. In this case, for example, if information such as how to distribute the new resource with respect to the volume configuration of the current virtual distributed file server system 4 is added, the volume can be selectively expanded, according to need.

[0066]

[Fourth Embodiment]

Fig. 6 is a block diagram illustrating the configuration of a computer network system to which the virtual distributed file server system according to the fourth embodiment of the present invention is applied, wherein like reference signs refer to like parts in Fig. 4.

[0067] In Fig. 6, reference sign 6 denotes a virtual distributed file server system corresponding to the virtual distributed file server system 4 in Fig. 4. The

characteristic point of the virtual distributed file server system 6 is that it includes each file load status information holding sections 16-1 to 16-n that hold the information of load status for each file (load status information for each file) held (in the storage devices 12-1 to 12-n) of the own server, in the file servers 10-1 to 10-n constituting the system 6. Therefore, the function of the virtual distributed file system 61, being the center of the virtual distributed file server system 6, is partially different from that of the virtual distributed file system 41 in Fig. 4. However, for the convenience sake, like reference signs (410-1 to 410-n) are used for the virtual distributed file modules for each file server 10-1 to 10-n in the virtual distributed file system 61. In Fig. 6, components (virtual distributed file interfaces, local file interfaces, and communication modules) in the virtual distributed file modules 410-1 to 410-n, and the mapping tables and the server information holding sections on the file servers 10-1 to 10-n are omitted.

[0068] The each file load status information holding section 16-i (i = 1 to n) has the data structure shown in Fig. 7(a), and holds the load status information for each file including the information indicating the load status for each file in the own server, the information indicating the attribute of the file (file attribute), and a replication flag. The file attribute indicates whether the corresponding file is the original or a replica (copy). When the corresponding file is the original, the replication flag indicates whether a replica thereof has already been generated on the other server side, that is, whether the replication has been already made.

[0069] In Fig. 6 is shown the situation in which a replica 612 of an optional file 611 in the storage device 12-1 included in the file server 10-1 is held by replication B1 in the storage device 12-n included in the file server 10-n.

[0070] The operation of the configuration shown in Fig. 6 will be explained. The virtual distributed file modules 410-1 to 410-n on the virtual distributed file system 61

refer to the each file load status information holding sections 16-1 to 16-n, for example, regularly. When having detected that a file having a load exceeding a first threshold exists in the files held by (the storage devices 12-1 to 12-n in) the own server, from the load status information for each file held by the holding sections 16-1 to 16-n, the modules 410-1 to 410-n perform the replication operation for generating a replica of the corresponding file asynchronously with respect to one of the other servers, for example, via the private communication path 5 by inter-server communication. Here, the load status of the file is the sum total of the number of requests in a request queue with respect to the file, or the size indicated by the requests in the waiting state of the file, and is updated every time a request is accepted and every time the request is processed. For the server to be replicated, for example, a server having the lowest load may be selected based on the server information held in the server information holding section (not shown).

[0071] When having performed the replication operation, the virtual distributed file modules 410-1 to 410-n sets the replication flag in the load status information of the corresponding file held by the each file load status information holding sections 16-1 to 16-n in the own server to a replication notification state. The virtual distributed file module of the server to be replicated adds the load status information of the corresponding replica to the each file load status information holding section of the own server.

[0072] Here, as shown in Fig. 6, it is assumed that replication B1 of the file 611 held by the file server 10-1 is performed with respect to the file server 10-n via the private communication path 5, and a replica 612 thereof is held in the storage device 12-n of the file server 10-n. In this case, the replication flag in the load status information of the file 612 held by the each file load status information holding section 16-1 in the file

server 10-1 is set to the state indicating that replication has been finished. New load status information relating to the replica 612 of the file 611 is added to the each file load status information holding section 16-n in the file server 10-n. The file attribute in the load status information indicates that the corresponding file is a replica (612) (of the file 611).

[0073] Thereafter, when there is a new read request of the file 611 from the client 2, (the virtual distributed file modules 410-1 in) the file server 10-1 holding the file 611 checks if the load of the file 611 exceeds a second threshold (second threshold < first threshold), and if the load exceeds the second threshold, the file server 10-1 does not respond to the request from the client 2. In this case, the file server 10-n having accepted the replication responds to the request from the client 2. Here, the file server 10-n does not have to consider whether the file server 10-1 responds thereto, and may respond to the client 2, so long as it has the replica 612 of the requested file 611.

[0074] In this manner, since the file server 10-n processes the new read request with respect to the file 611 from the client 2 by using the replica 612, in the file server 10-1 holding the file 611, the processing with respect to the read request of the file 611, which has been accepted before, progresses, and the load of the file 611 becomes not larger than the second threshold. Then, the virtual distributed file module 410-1 on the file server 10-1 transmits a request for deleting the replica 612 of the file 611 to the virtual distributed file module 410-n on the file server 10-n, for example, by inter-server communication via the private communication path 5.

[0075] The virtual distributed file module 410-n on the file server 10-n having received the request perfoms processing only for the already accepted request by using the replica 612, and thereafter, deletes the replica 612 and the corresponding load status

information. On the other hand, when there is a new read request with respect to the file 611 from the client 2, the virtual distributed file module 410-1 on the file server 10-1 responds thereto.

[0076] Since the read request with respect to the file 611 is accepted by the file server 10-n, according to the replication of the file 611 from the file server 10-1 to the file server 10-n, the load of the replica 612 of the file 611 in the file server 10-n may exceed the first threshold, before the load of the file 611 in the file server 10-1 becomes not larger than the second threshold.

[0077] Therefore, in such a case, the file server 10-n generates the next generation replica in another server by using the replica 612, that is, performs replication of the replica, and allows the server to handle the read request with respect to the file 611. For this purpose, as shown in Fig. 7(b), the load status information for each file held by the each file load status information holding section 16-i (i = 1 to n) may include the generation information of the file, in addition to the load status and the replication flag as shown in Fig. 7(a).

[0078] In this case, at a point in time when the load of the replica of a certain generation becomes not larger than the second threshold, the server having the replica can perform management such that the next generation replica held by a server, to which the replication has been performed, is deleted. At this time, when the server, to which deletion of the replica is requested, has generated the next generation replica in another server, the next generation replica can be also deleted. As for at least the load status information of the file relating to the replica, with regard to the same file (including the replica), a server holding the file having the lowest load may respond to the read request with respect to the file, by making the contents of information agree with each other between respective servers, as with the server information described

above.

[0079] Recently, there is an increase in data, which is basically for readout, and has a relatively large size and requires a certain degree of response (band guarantee according to a case), such as streaming data for video and audio, or contents of the World Wide Web (WWW). For these data, since accesses may be concentrated on a specific data (file), securing the response may be difficult. The configuration in Fig. 6 assumes such a situation, and when accesses are concentrated on a specific file, replication of the file is automatically performed, so that the access to the file can be distributed. This configuration can be used not only for load balancing, but also for backup of a file of great importance.

[0080]

[Fifth Embodiment]

Fig. 8 is a block diagram illustrating the configuration of a computer network system to which the virtual distributed file server system according to the fifth embodiment of the present invention is applied, wherein like reference signs refer to like parts in Fig. 4.

[0081] In Fig. 8, reference sign 8 denotes a virtual distributed file server system corresponding to the virtual distributed file server system 4 in Fig. 4. The characteristic point of this virtual distributed file server system 8 is that a network configuration is applied such that the file servers 10-1 to 10-n and the storage devices 12-1 to 12-n are connected with each other, for example, by a fiber channel arbitrated loop (FC-AL) 80, so that the respective file servers 10-1 to 10-n (as a host) can share the storage devices 12-1 to 12-n (as a target) (that is, multi-hosting is possible). Here, it should be noted that the private communication path 5 is not provided, different from the configuration shown in Fig. 4.

[0082] In the configuration shown in Fig. 8, the inter-server communication, performed via the private communication path 5 (by the communication modules 413-1 to 413-n in the virtual distributed file modules 410-1 to 410-n) in the configuration of Fig. 4, may be performed via the network 3 as in the configuration shown in Fig. 1 (Fig. 8 illustrates this state). The inter-server communication may be performed on the FC-AL 80 via an interface for connecting the storages in the file servers 10-1 to 10-n. In this case, the load on the network 3 can be reduced, in the same manner when the private communication path 5 is used.

[0083] According to the configuration of Fig. 8, since the storage devices 12-1 to 12-n can be directly seen from all file servers 10-1 to 10-n, the replication operation and load balancing as described in the fourth embodiment can be easily performed, by providing the each file load status information holding sections 16-1 to 16-n shown in Fig. 6 to the respective servers 10-1 to 10-n. The network (interface) capable of multi-hosting is not limited to the FC-AL 80, and may be a small computer system interface (SCSI) bus.

[0084]

[Effects of the Invention]

According to the present invention described above, clients can handle a plurality of file servers distributed on the network as a single server, and clients do not have to consider the number of servers and the connection state of the storage devices.

[Brief Description of the Drawings]

[Fig. 1] A block diagram illustrating the configuration of a computer network system to which a virtual distributed file server system according to the first embodiment of the

present invention is applied.

[Fig. 2] A diagram illustrating an example of data structure of a mapping table shown in Fig. 1.

[Fig. 3] A diagram illustrating an example of data structure of a server information holding section shown in Fig. 1.

[Fig. 4] A block diagram illustrating the configuration of a computer network system to which the virtual distributed file server system according to the second embodiment of the present invention is applied.

[Fig. 5] A block diagram illustrating the configuration of a computer network system to which the virtual distributed file server system according to the third embodiment of the present invention is applied.

[Fig. 6] A block diagram illustrating the configuration of a computer network system to which the virtual distributed file server system according to the fourth embodiment of the present invention is applied.

[Fig. 7] A diagram illustrating an example of data structure of an each file load status information holding section shown in Fig. 6.

[Fig. 8] A timing chart for explaining the operation of the embodiment.

[Description of Signs]

1, 4, 6, 8 ··· Virtual distributed file server system

2 ··· Client

3 ··· Network

5 ··· Private communication path

10-1 to 10-n ··· File server

11, 41, 61 ··· Virtual distributed file system

12-1 to 12-n ··· Storage device

13-1 to 13-n ··· Local file system

14-1 to 14-n ··· Mapping table

15-1 to 15-n ··· Server information holding section

16-1 to 16-n ··· Each file load status information holding section

80 ··· FC-AL (interface capable of multi-hosting)

110-1 to 110-n, 410-1 to 410-n ···Virtual distributed file module (Management module)

111-1 to 111-n ··· Virtual distributed file interface

112-1 to 112-n ··· Local file interface

113-1 to 113-n ··· Communication module

611 ··· File

612 ··· Replica (of the file 611)

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2001－51890
（P2001－51890A）
(43)公開日　平成13年２月23日(2001.2.23)

(51)Int.Cl.⁷	識別記号	ＦＩ	テーマコード（参考）
Ｇ０６Ｆ　12/00	５４５	Ｇ０６Ｆ　12/00	５４５Ａ　５Ｂ０８２
13/00	３５１	13/00	３５１Ｅ　５Ｂ０８９

審査請求　未請求　請求項の数７　ＯＬ　（全 14 頁）

(21)出願番号　特願平11－226494

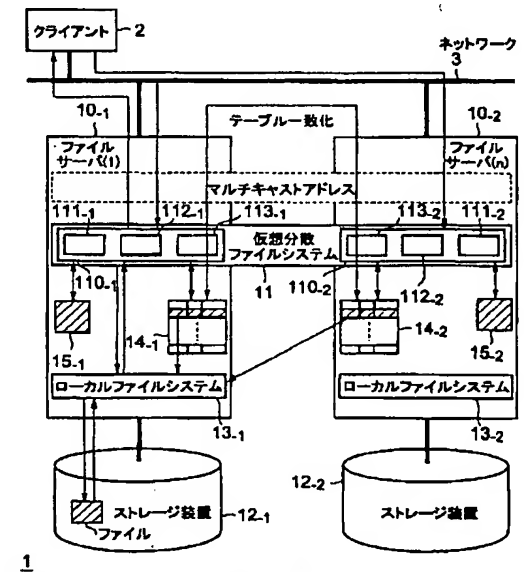
(22)出願日　平成11年８月10日(1999.8.10)

(71)出願人　000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地
(72)発明者　内堀　郁夫
東京都府中市東芝町１番地　株式会社東芝府中工場内
(72)発明者　高桑　正幸
東京都府中市東芝町１番地　株式会社東芝府中工場内
(74)代理人　100058479
弁理士　鈴江　武彦　（外６名）

最終頁に続く

(54)【発明の名称】　仮想分散ファイルサーバシステム

(57)【要約】
【課題】ネットワーク上に分散した複数のファイルサーバの台数やストレージ装置の接続状態をクライアントに意識させないで済むようにする。
【解決手段】ネットワーク３上に分散したサーバ１０-1，１０-2を備え、各サーバには、仮想分散ファイルシステム１１が分散して実装されている。このシステム１１を構成する、サーバ１０-1，１０-2上のモジュール１１０-1，１１０-2は、クライアント２からマルチキャストされたファイル操作要求を受け取ると、仮想分散ファイルシステム１１と全ローカルファイルシステム１３-1，１３-2とのマッピングテーブル１４-1，１４-2または全サーバのサーバ情報を保持するサーバ情報保持部１５-1，１５-2をもとに、自サーバが上記要求を処理可能な最適なサーバであるか否かを判断し、その判断結果に基づいて要求されたファイル操作を対応するサーバのローカルファイルシステムにより行わせる。



1

【特許請求の範囲】

【請求項１】　マルチキャスト可能なネットワーク上に分散した複数のファイルサーバを備えた仮想分散ファイルサーバシステムであって、

前記各ファイルサーバに分散して実装され、全ファイルサーバのファイルを統合的に管理する、実際のストレージ構成には非依存の仮想分散ファイルシステムと、

前記各ファイルサーバにそれぞれ独立して実装され、各サーバに固有のストレージ構成を管理するローカルファイルシステムと、

前記各ファイルサーバにそれぞれ設けられ、前記仮想分散ファイルシステムで統合的に管理される各ファイルについて、当該仮想分散ファイルサーバシステムとそのファイルを実際に管理する前記ローカルファイルシステムとの間のマッピングの情報を保持するマッピングテーブルとを具備し、

前記仮想分散ファイルシステムは、前記各ファイルサーバにそれぞれ設けられた管理モジュールから構成され、

前記各管理モジュールは、クライアントからマルチキャストされたファイル操作要求を共通に受け取り、当該要求に応じて自サーバの前記マッピングテーブルを参照することで、自サーバが当該要求を処理可能な最適なサーバであるか否かを判断し、最適なサーバであると判断した場合だけ、要求されたファイル操作を対応するサーバの前記ローカルファイルシステムにより行わせるように構成されていることを特徴とする仮想分散ファイルサーバシステム。

【請求項２】　マルチキャスト可能なネットワーク上に分散した複数のファイルサーバを備えた仮想分散ファイルサーバシステムであって、

前記各ファイルサーバに分散して実装され、全ファイルサーバのファイルを統合的に管理する、実際のストレージ構成には非依存の仮想分散ファイルシステムと、

前記各ファイルサーバにそれぞれ独立して実装され、各サーバに固有のストレージ構成を管理するローカルファイルシステムと、

前記各ファイルサーバにそれぞれ設けられ、前記仮想分散ファイルシステムで統合的に管理される各ファイルについて、当該仮想分散ファイルサーバシステムとそのファイルを実際に管理する前記ローカルファイルシステムとの間のマッピングの情報を保持するマッピングテーブルと、

前記各ファイルサーバにそれぞれ設けられ、全ての前記ファイルサーバについて、そのサーバのストレージ装置の空き容量を示す情報、及びそのサーバの負荷状況を示す情報の少なくとも一方を含むサーバ情報を保持するサーバ情報保持手段とを具備し、

前記仮想分散ファイルシステムは、前記各ファイルサーバにそれぞれ設けられた管理モジュールから構成され、

前記各管理モジュールは、クライアントからマルチキャストされたファイル操作要求を共通に受け取り、当該要求に応じて自サーバの前記マッピングテーブルまたは前記サーバ情報保持手段を参照することで、自サーバが当該要求を処理可能な最適なサーバであるか否かを判断し、最適なサーバであると判断した場合だけ、要求されたファイル操作を対応するサーバの前記ローカルファイルシステムにより行わせるように構成されていることを特徴とする仮想分散ファイルサーバシステム。

【請求項３】　前記管理モジュールは、前記ファイル操作要求がファイル読み出し要求またはファイル書き込み要求の場合には、自サーバの前記マッピングテーブルを参照し、該当するファイルが自サーバの前記ローカルファイルシステムの管理下にあるか否かにより、自サーバが前記要求を処理可能な最適なサーバであるか否かを判断することを特徴とする請求項１または請求項２記載の仮想分散ファイルサーバシステム。

【請求項４】　前記管理モジュールは、前記ファイル操作要求がファイルの新規作成要求の場合には、自サーバの前記サーバ情報保持手段を参照し、全ての前記サーバの各々について、そのサーバのストレージ装置の空き容量、またはそのサーバの負荷状況を比較することで、自サーバが前記要求を処理可能な最適なサーバであるか否かを判断することを特徴とする請求項２記載の仮想分散ファイルサーバシステム。

【請求項５】　前記管理モジュールは、全ての前記ファイルサーバの前記マッピングテーブルの内容を一致化するために、自サーバの前記マッピングテーブルの情報と他のサーバの前記マッピングテーブルの情報とをサーバ間通信により交換することを特徴とする請求項１記載の仮想分散ファイルサーバシステム。

【請求項６】　前記管理モジュールは、全ての前記ファイルサーバの前記マッピングテーブルの内容を一致化するために、自サーバの前記マッピングテーブルの情報と他のサーバの前記マッピングテーブルの情報とをサーバ間通信により交換する一方、全ての前記ファイルサーバの前記サーバ情報保持手段の内容を一致化するために、自サーバの前記サーバ情報保持手段の情報と他のサーバの前記サーバ情報保持手段の情報とをサーバ間通信により交換することを特徴とする請求項２記載の仮想分散ファイルサーバシステム。

【請求項７】　前記各ファイルサーバにそれぞれ設けられ、そのサーバの管理下にある各ファイル別の負荷状況を示す情報を保持するファイル別負荷状況情報保持手段を更に具備し、

前記管理モジュールは、自サーバの前記ファイル別負荷状況情報保持手段に保持されている情報から第１の閾値を超えた負荷のファイルを検出して、他の任意のファイルサーバに対してサーバ間通信により当該ファイルのレプリケーションを行い、クライアントからマルチキャストされた当該ファイルの読み出し要求があった場合、当

該要求に対する処理をレプリケーション側に任せるようにしたことを特徴とする請求項１または請求項２記載の仮想分散ファイルサーバシステム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、コンピュータ・ネットワークシステムにおけるファイルサーバシステムに係り、特にネットワーク上に接続された複数のファイルサーバを協調動作させて、クライアントからは単一のサーバとして機能させる仮想分散ファイルサーバシステムに関する。

【０００２】

【従来の技術】今日のコンピュータ・ネットワークシステムにおいては、ネットワークに接続された異なるコンピュータ間でファイルを共有することが一般的に行われている。こうした環境下では、特定のコンピュータに大規模なストレージを接続して、ファイルサーバとして運用したり、最近ではＮＡＳ（Network Attached Storage）と呼ばれる、ファイルサーバ専用機を接続する等のシステム構成をとることが多い。

【０００３】ファイルサーバを使用する環境（ファイルサーバシステム）では、サーバのストレージ容量が不足した場合には、サーバ側に物理的・性能的に拡張性があれば、新たにディスク装置等（のストレージ装置）を増設することで対処できる。このときクライアントからは、新たなボリュームをマウントして使用するといった形態になる。また、サーバの拡張性が限界に達していれば、サーバ自体を増設することになる。このときクライアントからは、増設したサーバを意識した上で新たなボリュームをマウントして使用するといった形態になる。

【０００４】

【発明が解決しようとする課題】上記したコンピュータ・ネットワークシステムにおいてファイルサーバを利用してファイル共有を行う場合、クライアント側からは、ファイルサーバ側のボリューム構成がそのまま見えてしまうのが一般的である。例えばサーバ側でディスク装置を増設した場合には、クライアント側は新たなボリュームを認識した上で、マウントしなければならない。或いはサーバ自体を増設した場合には、増設したサーバの運用ポリシーを決定、もしくはシステム設定・管理等の煩雑な作業が発生する上、クライアント側でも、新たなサーバを認識した上で、新たなボリュームをマウントしなければならない。

【０００５】このように従来のファイルサーバを用いたファイル共有システム（ファイルサーバシステム）では、ディスク装置（ストレージ装置）の増設、或いはサーバの増設が必要な場合、サーバ側、クライアント側のいずれにも、新たな設定・管理のために多大なコストが発生するという問題があった。更に、ストレージの利用形態よっては、特定のファイルシステムをそのまま容量

だけ拡張したい場合もあり、単にストレージ装置やサーバを増設するだけでは解決しないケースもあった。

【０００６】本発明は上記事情を考慮してなされたものでその目的は、ネットワーク上に分散した複数のファイルサーバを、クライアントからは単一のサーバとして扱うことができ、サーバ台数やストレージ装置の接続状態をクライアントに意識させない仮想分散ファイルサーバシステムを提供することにある。

【０００７】

【課題を解決するための手段】本発明は、マルチキャスト可能なネットワークに接続された複数のファイルサーバに分散して実装され、全ファイルサーバのファイルを統合的に管理する、実際のストレージ構成には非依存の仮想分散ファイルシステムと、各ファイルサーバにそれぞれ独立して実装され、各サーバに固有のストレージ構成を管理するローカルファイルシステムと、前記各ファイルサーバにそれぞれ設けられ、上記各ファイルについて、仮想分散ファイルサーバシステムとそのファイルを実際に管理するローカルファイルシステムとの間のマッピングの情報（例えば、仮想分散ファイルサーバシステムで管理され、クライアントから見える仮想的なパスと、ローカルファイルシステムで管理され、クライアントから見えない物理的な所在とを対応付けた情報）を保持するマッピングテーブルとを備えると共に、上記仮想分散ファイルシステムを、各ファイルサーバにそれぞれ設けられた管理モジュールであって、クライアントからマルチキャストされたファイル操作要求を共通に受け取り、当該要求に応じて自サーバのマッピングテーブルを参照することで、自サーバが当該要求を処理可能な最適なサーバであるか否かを判断し、最適なサーバであると判断した場合だけ、要求されたファイル操作を対応するサーバのローカルファイルシステムにより行わせる管理モジュールにより構成することを特徴とする。

【０００８】ここで、各ファイルサーバ上に、全ファイルサーバについて、そのサーバのストレージ装置の空き容量を示す情報、及びそのサーバの負荷状況を示す情報の少なくとも一方を含むサーバ情報を保持するサーバ情報保持手段を更に設け、上記各管理モジュールでは、クライアントからマルチキャストされたファイル操作要求を受け取った場合に、当該要求に応じて自サーバのマッピングテーブルまたはサーバ情報保持手段を参照することで、自サーバが当該要求を処理可能な最適なサーバであるか否かを判断する構成としてもよい。

【０００９】このような構成においては、クライアントから特定のファイルサーバを意識することなくマルチキャストされたファイル操作要求は、仮想分散ファイルサーバシステムを構成する各ファイルサーバ上の管理モジュールで共通に受け取られ、その要求に応じて対応するサーバ（自サーバ）のマッピングテーブルまたはサーバ情報保持手段が参照される。そして、この参照の結果、

自サーバが上記要求を処理可能な最適なサーバであるか否かが判断され、最適なサーバであると判断した唯一のサーバ（上の管理モジュール）だけが、要求されたファイル操作を自サーバのローカルファイルシステムにより行わせる。

【００１０】このように、要求元のクライアントからは、ネットワーク上に分散した複数のファイルサーバを単一のサーバとして扱うことができ、サーバ台数やストレージ装置の接続状態を意識する必要がない。

【００１１】ここで、上記管理モジュールで、自サーバが最適なサーバであるか否かを判断するためのアルゴリズムとして、以下の第１乃至第４のアルゴリズム（判断手法）のいずれかを適用するとよい。

【００１２】第１のアルゴリズムは、ファイル操作要求がファイル読み出し要求またはファイル書き込み要求の場合に適用されるもので、自サーバのマッピングテーブルの情報に基づいて、該当するファイルが自サーバのローカルファイルシステムの管理下にあるか否かにより判断する手法である。

【００１３】第２のアルゴリズムは、ファイル操作要求がファイルの新規作成要求の場合に適用されるもので、自サーバのサーバ情報保持手段の情報に基づいて、全てのサーバの各々について、そのサーバのストレージ装置の空き容量（空き記憶容量）、またはそのサーバの負荷状況を比較することで判断する（例えば、自サーバの空き容量が最も大きい場合、或いは自サーバの負荷が最も低い場合に上記最適サーバと判断する）手法である。

【００１４】第３のアルゴリズムも、ファイル操作要求がファイルの新規作成要求の場合に適用されるもので、自サーバのマッピングテーブルの情報に基づいて、全てのサーバの各々について対応するストレージ装置上に確保可能な連続領域を求め、その連続領域のサイズを比較することで判断する（例えば、自サーバのストレージ装置上に確保可能な連続領域のサイズが最も大きい場合に上記最適サーバと判断する）手法である。

【００１５】第４のアルゴリズムも、ファイル操作要求がファイルの新規作成要求の場合に適用されるもので、全てのサーバの各々について、そのサーバのストレージ装置の空き容量、そのサーバの負荷、及び当該ストレージ装置上に確保可能な連続領域の少なくとも２つを求め、その求めた少なくとも２つの情報を複合条件として比較することで判断する手法である。

【００１６】以上の第１乃至第４のアルゴリズムのいずれか１つを適用することで、クライアントから特定のファイルサーバを意識することなくマルチキャストされたファイル操作要求を各サーバが共通に受け取っても、その要求されたファイル操作を行うのに最適なサーバであるか否かを、その都度相互に通信を行うことなく、そのサーバ自身で自律的に判断することができる。

【００１７】ここで、上記各管理モジュールに、全ての

ファイルサーバのマッピングテーブルの内容を一致化するために、自サーバのマッピングテーブルの情報と他サーバのマッピングテーブルの情報とをサーバ間通信により交換する機能（通信モジュール）を持たせるとよい。また、マッピングテーブルに加えてサーバ情報保持手段を各サーバ上に備えた構成では、各管理モジュール（内の通信モジュール）に、全てのファイルサーバのサーバ情報保持手段の内容を一致化するために、自サーバのサーバ情報保持手段の情報と他のサーバのサーバ情報保持手段の情報とをサーバ間通信により交換する機能を更に持たせるとよい。

【００１８】また、マッピングテーブルの一致化のためには、自サーバのローカルファイルシステムで実際に管理されるファイル構成が変更された場合に、その変更された情報（マッピング情報）をサーバ間通信により他の全サーバに送信するのが効率的である。同様に、サーバ情報保持手段の内容の一致化のためには、自サーバのサーバ情報を定期的に更新し、その都度、その更新されたサーバ情報をサーバ間通信により他の全サーバに送信するのが効率的である。

【００１９】また本発明は、上記仮想分散ファイルサーバシステムにサーバが動的に増設された場合に、そのサーバの管理モジュールで以下の第１乃至第４の処理を行うようにしたことをも特徴とする。まず、第１の処理では、サーバ間通信により他の全てのサーバに対してマッピングテーブル及びサーバ情報保持手段の更新を禁止するロック設定を行い、次の第２の処理では、サーバ間通信により他の任意のサーバからマッピングテーブルサーバ情報保持手段の内容を自サーバにコピーし、次の第３の処理では、自サーバのサーバ情報保持手段に自サーバのサーバ情報を追加し、次の第４の処理では、サーバ間通信により自サーバのーバ情報を他の全てのサーバのサーバ情報保持手段に反映させて全サーバのサーバ情報保持手段の一致化を図り、しかる後に上記ロック設定を解除する。

【００２０】このようなサーバ増設時の一連の動作により、動的にサーバ台数を拡張できる。しかもクライアントは、サーバ台数の拡張を意識することなく、増設されたサーバを利用することができる。

【００２１】また本発明は、各ファイルサーバに、そのサーバの管理下にある各ファイル別の負荷状況を示す情報を保持するファイル別負荷状況情報保持手段を付加し、各サーバの管理モジュールにおいて、自サーバのファイル別負荷状況情報保持手段に保持されている情報から第１の閾値を超えた負荷のファイルを検出して、他の任意のファイルサーバに対してサーバ間通信により当該ファイルのレプリケーションを行い、クライアントからマルチキャストされた当該ファイルの読み出し要求があった場合、当該要求に対する処理をレプリケーション側に任せるようにしたことをも特徴とする。

【００２２】このような構成においては、自律的な負荷分散が可能となる。ここで、上記検出したファイルがレプリケーションされたファイルである場合にも、他の任意のファイルサーバに対してサーバ間通信により当該ファイルのレプリケーションを行い、クライアントからマルチキャストされた当該ファイルの読み出し要求があった場合に、当該要求に対する処理を新たなレプリケーション側に任せることにより、自律的な負荷分散がより広範囲に効果的に行える。また、他サーバへのレプリケーションの対象となったファイルの負荷が前記第１の閾値より低い第２の閾値以下となった場合に、そのレプリケーションを行ったサーバ上の管理モジュールから当該他サーバに対してサーバ間通信により対応するファイルの消去を要求し、クライアントからマルチキャストされた当該ファイルの読み出し要求があった場合、当該要求に対する処理を自身が行うことにより、動的な負荷分散が可能となる。

【００２３】さて、上記一致化のためのサーバ間通信（マッピング情報またはサーバ情報の通信）、更には上記レプリケーションのためのサーバ間通信には、上記ネットワークを用いることが可能である。しかし、各ファイルサーバを相互接続する専用の通信路（プライベート通信路）を上記ネットワークから独立に設け、当該通信路を用いてサーバ間通信を行う構成とするとよい。この場合、サーバ間通信のためにネットワークのスループットが悪化するのを防止できる。

【００２４】また本発明は、各ファイルサーバ及び当該サーバのストレージ装置を相互接続するマルチホストが可能なインタフェースを更に備えると共に、上記各管理モジュールによる上記サーバ間通信を当該インタフェースを介して行うようにしたことをも特徴とする。

【００２５】このような構成においては、上記各ストレージ装置を上記インタフェースによって各サーバ間で共有し、上記一致化のためのサーバ間通信、更には動的なファイルのレプリケーションのためのサーバ間通信が上記インタフェースを通して行われるため、自律的な負荷分散が効果的に実現される。

【００２６】なお、本発明は方法に係る発明としても成立する。

【００２７】

【発明の実施の形態】以下、本発明の実施の形態につき図面を参照して説明する。

【００２８】［第１の実施形態］図１は本発明の第１の実施形態に係る仮想分散ファイルサーバシステムを適用するコンピュータ・ネットワークシステムの構成を示すブロック図である。

【００２９】同図において、１は仮想分散ファイルサーバシステム、２は同ファイルサーバシステム１にファイルサービスを要求するクライアント（クライアントコンピュータ）である。仮想分散ファイルサーバシステム１

は、ネットワーク３上に分散配置された複数、例えば２台のファイルサーバ（サーバコンピュータ）１０-1，１０-2を用いて実現される。なお、図ではクライアント２は便宜的に１台だけが示されているが、複数存在するのが一般的である。

【００３０】１１は仮想分散ファイルサーバシステム１の中心をなす仮想分散ファイルシステムであり、各ファイルサーバ１０-1，１０-2に分散して実装されている。この仮想分散ファイルシステム１１は、全ファイルサーバ１０-1，１０-2のファイルを統合的に管理し、ファイルサーバ１０-1，１０-2それぞれの実際のボリューム構成（ストレージ構成）には依存しない、仮想的なファイルシステムをクライアント２に対して提供するものである。

【００３１】仮想分散ファイルシステム１１は、各ファイルサーバ１０-1，１０-2に分散して実装された仮想分散ファイルモジュール１１０-1，１１０-2を有している。仮想分散ファイルモジュール１１０-1，１１０-2は、ファイルサーバ１０-1，１０-2上でクライアント２からの要求を分散して処理しつつ、クライアント２に対しては仮想的に１つのファイルシステムとして見せるための管理モジュールである。仮想分散ファイルモジュール１１０-1，１１０-2は、当該モジュール１１０-1，１１０-2の中心をなし、クライアント２からの要求を処理する仮想分散ファイルインタフェース１１１-1，１１１-2と、後述するローカルファイルシステム１３-1，１３-2とのインタフェース（ローカルファイルインタフェース）１１２-1，１１２-2と、当該モジュール１１０-1，１１０-2間での通信（後述するマッピングテーブル１４-1，１４-2の情報、及びサーバ情報の一致化のための通信に代表されるサーバ間通信）を行う通信モジュール１１３-1，１１３-2とを持つ。

【００３２】ファイルサーバ１０-1，１０-2はネットワーク３を介してクライアント２と接続されている。ファイルサーバ１０-1，１０-2は、仮想分散ファイルシステム１１の他に、それぞれ当該サーバ１０-1，１０-2に接続されたディスク装置などのストレージ装置１２-1，１２-2（実際のストレージ構成）を管理するローカルなファイルシステム（ローカルファイルシステム）１３-1，１３-2を実装している。

【００３３】ファイルサーバ１０-1，１０-2には、仮想分散ファイルシステム１１とローカルファイルシステム１３-1，１３-2とを対応付ける同一内容のマッピングテーブル１４-1，１４-2が設けられている。このテーブル１４-i（ｉ＝１，２）のデータ構造を図２に示す。

【００３４】テーブル１４-iの各エントリは、仮想分散ファイルシステム１が管理するファイルのファイル名の登録フィールド（ファイル名フィールド）１４１と、当該ファイルの仮想分散ファイルシステム１上の論理的な所在を表す（クライアント２から見える）パス（仮想パ

ス）の登録フィールド（仮想パスフィールド）１４２、当該ファイルのストレージ上の物理的な所在位置を表す（クライアント２から見えない）所在位置情報の登録フィールド（所在位置情報フィールド）１４３、当該ファイルへのアクセス権（許可／禁止）を管理するためのパーミッション情報の登録フィールド（パーミッション情報フィールド）１４４、及びその他の各種属性の登録フィールド１４５を有している。

【００３５】仮想分散ファイルシステム１１（上の仮想分散ファイルモジュール１１０-i）は、このようなデータ構造のマッピングテーブル１４-iを参照することにより、例えばあるファイルがファイルサーバ１０-1，１０-2のいずれにあるか等の所在情報を得ることができる他、パーミッション等、必要に応じてファイルの属性を得ることができる。

【００３６】ファイルサーバ１０-1，１０-2には更に、同一内容のサーバ情報保持部１５-1，１５-2が設けられている。サーバ情報保持部１５-i（ｉ＝１，２）は、図３に示すように、仮想分散ファイルサーバシステム１を構成する全てのファイルサーバ１０-1，１０-2の（ストレージ装置１２-1，１２-2の）空き記憶容量を示す情報（リソース情報）、及び負荷状況を示す情報を含むサーバ情報を保持するのに用いられる。

【００３７】次に図１の構成の動作を説明する。本実施形態では、クライアント２からは、各ファイルサーバ１０-1，１０-2のローカルファイルシステム１３-1，１３-2ではなくて、仮想分散ファイルシステム１１がマウントされているように見えている。そこでクライアント２は、何らかのファイル操作要求が発生した場合、仮想分散ファイルシステム１１が実装されている全ファイルサーバ１０-1，１０-2に対して同一の要求を発行する。この場合、例えばＩＰ（Internet Protocol）マルチキャストを使用する等の手法によれば、クライアント２側はファイルサーバの台数を意識することなく要求の発行が可能である。

【００３８】ファイルサーバ１０-1，１０-2は、クライアント２からの要求を受け取ると、当該要求を仮想分散ファイルシステム１１内の自サーバに対応した仮想分散ファイルモジュール１１０-1，１１０-2に渡す。すると、モジュール１１０-1，１１０-2（内の仮想分散ファイルインタフェース１１１-1，１１１-2）は、その要求がファイルの読み出し要求もしくは書き込み（更新）要求であるか、または新規ファイルの作成要求もしくはディレクトリの作成要求であるか、その要求種別を判別する。

【００３９】ここで、クライアント２からのファイル操作要求がファイルの読み出し要求もしくは書き込み要求であるものとする。この要求には、要求の対象となるファイルのファイル名と、当該ファイルの仮想分散ファイルシステム１１上のパス（仮想パス）が付されている。

【００４０】仮想分散ファイルモジュール１１０-1，１１０-2（内の仮想分散ファイルインタフェース１１１-1，１１１-2）は、クライアント２からのファイル操作要求がファイルの読み出し要求もしくは書き込み要求の場合、要求されたファイルのファイル名及び仮想パスにより自サーバ内のマッピングテーブル１４-1，１４-2を参照し、当該ファイル名及び仮想パスを持つテーブル１４-1，１４-2内エントリ中の所在位置情報フィールド１４３の登録情報から、操作要求のあったファイルが自サーバ内（自サーバに接続されたストレージ装置１２-1，１２-2）に保持されているか否かを調べる。

【００４１】もし、要求されたファイルが自サーバ内に保持されている場合には、仮想分散ファイルモジュール１１０-1，１１０-2（内の仮想分散ファイルインタフェース１１１-1，１１１-2）は、ローカルファイルインタフェース１１２-1，１１２-2により自サーバ内のローカルファイルシステム１３-1，１３-2を介して実際のファイルにアクセスし、クライアント２に応答を返す。一方、操作要求のあったファイルが自サーバ内になかった場合には、他のサーバが応答するものと見なして応答しない。

【００４２】これに対し、クライアント２からの要求が新規ファイルの作成、或いはディレクトリの作成であった場合には、仮想分散ファイルモジュール１１０-1，１１０-2は、（マッピングテーブル１４-1，１４-2ではなくて）サーバ情報保持部１５-1，１５-2を参照する。そして、サーバ情報保持部１５-1，１５-2に保持されている全サーバのサーバ情報をもとに、所定のアルゴリズムに従い、いずれか１つのサーバ１０-i（ｉは１または２）上の仮想分散ファイルモジュール１１０-iだけが、仮想分散ファイルインタフェース１１１-iによりクライアント２からの要求を受け付ける。具体的には、サーバ１０-i上の仮想分散ファイルモジュール１１０-iは、全サーバのサーバ情報の示す空き記憶容量を比較し、自サーバ１０-i（のストレージ装置１２-i）の空き記憶容量が最も大きいと判定できる場合に、クライアント２からの要求を受け付けるものとする。この場合、必ずしもサーバ情報中に負荷状況の情報を持たせる必要はない。

【００４３】なお、全サーバのサーバ情報の示す負荷を比較し、自サーバの負荷が最も低い場合にクライアント２からの要求を受け付けるようにしてもよい。この場合、必ずしもサーバ情報中に対応するサーバの空き記憶容量の情報を持たせる必要はない。

【００４４】この他に、マッピングテーブル１４-iの各ファイル毎のマッピング情報から（つまり各ストレージ装置１２-1，１２-2の領域の使用状況から）、各ストレージ装置１２-1，１２-2上に確保可能な連続領域を求め、必要なサイズ以上の連続領域が確保でき、且つそのサイズが最も大きいストレージ装置が自サーバのストレージ装置１２-iの場合に、クライアント２からの要求を

受け付けるようにしてもよい。この場合、サーバ情報保持部１５-1，１５-2は必ずしも必要でない。

【００４５】更に、空き記憶容量と負荷状況と確保できる連続領域のサイズの少なくとも２つを条件（複合条件）として評価値を求め、自サーバが要求を受け付ける最適なサーバであるか否かを判断ようにしてもよい。

【００４６】さて、ファイルサーバ１０-i上の仮想分散ファイルモジュール１１０-iでは、仮想分散ファイルインタフェース１１１-iによりクライアント２からの要求を受け付けると、要求された新規ファイルの作成、或いはディレクトリの作成を、ローカルファイルインタフェース１１２-iを介してローカルファイルシステム１３-iにより行い、マッピングテーブル１４-1，１４-2に該当するエントリ情報を登録する。

【００４７】新規ファイルの作成、或いはディレクトリの作成が完了した後は、ファイルサーバ１０-i上の仮想分散ファイルモジュール１１０-iでは、自サーバ上のマッピングテーブル１４-iに登録した新たなエントリ情報を通信モジュール１１３-iによりネットワーク３を介して他の全てのサーバ１０-j（ｊは１または２、但しｊ≠ｉ）上の仮想分散ファイルモジュール１１０-ｊに送る。仮想分散ファイルモジュール１１０-ｊ（内の仮想分散ファイルインタフェース１１１-j）は、仮想分散ファイルモジュール１１０-iから送られたマッピングテーブル１４-iのエントリ情報を通信モジュール１１３-jを介して受け取る。そしてモジュール１１０-ｊ（内のインタフェース１１１-j）は、受け取った他サーバ１０-iのマッピングテーブル１４-iのエントリ情報を自サーバ内のマッピングテーブル１４-jに登録する。このように、ファイルサーバ１０-1，１１０-2上の仮想分散ファイルモジュール１１０-1，１１０-2が相互にマッピングテーブル１４-1，１４-2の新規登録されたエントリ情報（更には更新されたエントリ情報）を交換し合うことで、当該マッピングテーブル１４-1，１４-2の内容の一致化を図ることができる。

【００４８】また、ファイルサーバ１０-1，１０-2上の仮想分散ファイルモジュール１１０-1，１１０-2は、自サーバ上のサーバ情報保持部１５-1，１５-2に保持されている各サーバのサーバ情報のうち、自サーバのサーバ情報（空き記憶容量、及び負荷状況）を定期的に更新すると共に、その更新したサーバ情報を（通信モジュール１１３-1，１１３-2により）ネットワーク３を介して他の全てのサーバ（上の仮想分散ファイルモジュール１１０-2，１１０-1）に定期的に送ることで、各ファイルサーバ１０-1，１０-2のサーバ情報保持部１５-1，１５-2の内容の一致化を図る。つまり仮想分散ファイルモジュール１１０-1，１１０-2は定期的にサーバ情報を交換し合うことで一致化を図る。

【００４９】以上の動作によって、ファイルサーバ１０-1，１０-2を自律的に分散・協調動作させることができ、クライアント２には実際にはファイルサーバが２台（複数台）あることを意識させずに、仮想的なファイルサーバを提供することができる。

【００５０】なお、図１のシステムの例ではサーバが２台である場合について説明したが、サーバが３台以上であっても同様の仕組みによって、仮想的なファイルサーバを提供することができる。

【００５１】［第２の実施形態］図４は本発明の第２の実施形態に係る仮想分散ファイルサーバシステムを適用するコンピュータ・ネットワークシステムの構成を示すブロック図であり、図１と同一部分には同一符号を付してある。

【００５２】図４において、仮想分散ファイルサーバシステム４の中心をなす仮想分散ファイルシステム４１は、ｎ台のファイルサーバ１０-1～１０-nに分散して実装されている。この仮想分散ファイルシステム４１は、図１中の仮想分散ファイルシステム１１と同様に、全ファイルサーバ１０-1～１０-nのファイルを統合的に管理し、各ファイルサーバ１０-1～１０-nそれぞれの実際のボリューム構成には依存しない、仮想的なファイルシステムをクライアント２に対して提供している。仮想分散ファイルシステム４１は、各ファイルサーバ１０-1～１０-n上でクライアント２からの要求を処理する仮想分散ファイルモジュール４１０-1～４１０-nを有している。モジュール４１０-1～４１０-nは、図１中のモジュール１１０-1，１１０-2と同様の構成である、仮想分散ファイルインタフェース４１１-1～４１１-nと、ローカルファイルインタフェース４１２-1～４１２-nと、通信モジュール４１３-1～４１３-nとを持つ。但し、本実施形態における通信モジュール４１３-1～４１３-nは、図１中の通信モジュール１１３-1，１１３-2と異なり、後述するプライベート通信路５を介して通信を行うように構成されている。

【００５３】ファイルサーバ１０-1～１０-nはネットワーク３を介してクライアント２と接続されている。ファイルサーバ１０-1～１０-nは、仮想分散ファイルシステム１１の他に、それぞれ当該サーバ１０-1～１０-nに接続されたストレージ装置１２-1～１２-2を管理するローカルなファイルシステム（ローカルファイルシステム）１３-1～１３-2を実装している。ファイルサーバ１０-1～１０-nには、マッピングテーブル１４-1～１４-nと、サーバ情報保持部１５-1～１５-2とが設けられている。

【００５４】図４の構成の仮想分散ファイルサーバシステム４の特徴は、図１の構成の仮想分散ファイルサーバシステム１と異なって、システムを構成するファイルサーバの台数がｎ台である点と、そのｎ台のファイルサーバ１０-1～１０-nがネットワーク３とは別のプライベートな通信路５によっても相互接続されている点である。このプライベート通信路５は、例えばイーサネット、或いはファイバチャネル（Fibre Channnel）等であるが、

物理層に関しては特定しない。またトポロジに関しても、図４の例ではバス型を想定しているが、ループやスイッチであってもよい。

【００５５】図４の構成において、ファイルサーバ１０-1～１０-nが（仮想分散ファイルシステム４１内の仮想分散ファイルモジュール４１０-1～４１０-nにより）分散・協調動作を行うためには、前記第１の実施形態での動作説明から類推されるように、マッピングテーブル１４-1～１４-n、及びサーバ情報保持部１５-1～１５-nの内容を、各サーバ１０-1～１０-n間で常に一致化させておく必要がある。しかし、サーバ１０-1～１０-n間の情報一致化を、前記第１の実施形態と同様にネットワーク３を介して行うのでは、仮想分散ファイルサーバシステム４を構成するファイルサーバの台数が増加した場合には、その情報一致化の（ためのサーバ間通信の）トラフィックが増加し、ネットワーク３上のスループットを悪化させることになる。

【００５６】そこで本実施形態（第２の実施形態）では、図４の構成のように、各ファイルサーバ１０-1～１０-n間にサーバ間の情報交換専用のプライベート通信路５を設け、仮想分散ファイルシステム４１内の仮想分散ファイルモジュール４１０-1～４１０-nで通信モジュール４１３-1～４１３-nにより行われるサーバ間通信に、即ちマッピングテーブル１４-1～１４-n、及びサーバ情報保持部１５-1～１５-nの内容を一致化するためのサーバ間通信に、この通信路５を使用するようにしている。

【００５７】このように本実施形態では、マッピングテーブル１４-1～１４-n、及びサーバ情報保持部１５-1～１５-nの内容の一致化のためのサーバ間通信に、ネットワーク３でなくてプライベート通信路５を用いることにより、ネットワーク３の負荷の軽減を図ることができる。

【００５８】［第３の実施形態］以上に述べた第１、第２の実施形態では、複数のファイルサーバを分散・協調動作させる仮想分散ファイルサーバシステムの構成例を示した。この第１、第２の実施形態で参照した図１、図４の構成は、特定のサーバ台数における静的な例である。しかし、サーバ台数については、変更可能な構成とすることが好ましい。

【００５９】そこで、仮想分散ファイルサーバシステムを構成するサーバ台数を動的に拡張可能とした本発明の第３の実施形態について図面を参照して説明する。図５は本発明の第３の実施形態に係る仮想分散ファイルサーバシステムを適用するコンピュータ・ネットワークシステムの構成を示すブロック図であり、図４と同一部分には同一符号を付してある。

【００６０】まず、図４に示した仮想分散ファイルサーバシステム４、即ちｎ台のファイルサーバ１０-1～１０-nで構成される仮想分散ファイルサーバシステム４に、図５（ａ）に示すように、新たなファイルサーバ１０-(n+1)を追加するものとする。

【００６１】この場合、追加されたファイルサーバ１０-(n+1)にも分散されている仮想分散ファイルシステム４１上の仮想分散ファイルモジュール４１０-(n+1)は、既に仮想分散ファイルサーバシステム４を構成しているファイルサーバ１０-1～１０-nに対し、図５（ａ）において符号Ａ１で示すように、（例えば図示せぬプライベート通信路を介しての）サーバ間通信により、マッピングテーブル１４-1～１４-nのエントリ情報及びサーバ情報保持部１５-1～１５-nのサーバ情報（各サーバのリソース情報及び負荷状況を含む）の更新をロックする。

【００６２】その上で、追加されたサーバ１０-(n+1)上のモジュール４１０-(n+1)は、他のファイルサーバ１０-1～１０-nのうちのいずれかのサーバ、例えばファイルサーバ１０-1から、図５（ｂ）において符号Ａ２で示すように、マッピングテーブル１４-1及びサーバ情報保持部１５-1の全情報を、サーバ間通信により自サーバ内のマッピングテーブル１４-(n+1)及びサーバ情報保持部１５-(n+1)にコピーする。

【００６３】次に、追加されたファイルサーバ１０-(n+1)上のモジュール４１０-(n+1)は、コピー後のサーバ情報保持部１５-(n+1)に対し、図５（ｃ）において符号Ａ３で示すように、自サーバのリソース及び負荷状況を示すサーバ情報を追加する。

【００６４】しかる後にファイルサーバ１０-(n+1)上のモジュール４１０-(n+1)は、図５（ｄ）において符号Ａ４で示すように、サーバ間通信により他の全ファイルサーバ１０-1～１０-nに対してサーバ情報の一致化要求を発行し、その後にロックを解除する。

【００６５】以上の一連の動作により、既に構築されている仮想分散ファイルサーバシステム４に対して、動的に新たなサーバ（ファイルサーバ１０-(n+1)）を追加することができる。この場合、例えば現在の仮想分散ファイルサーバシステム４のボリューム構成に対し、新規リソースをどのように振り分けるか、といった情報を付加すれば、必要に応じてボリュームを選択的に拡張することも可能である。

【００６６】［第４の実施形態］図６は本発明の第４の実施形態に係る仮想分散ファイルサーバシステムを適用するコンピュータ・ネットワークシステムの構成を示すブロック図であり、図４と同一部分には同一符号を付してある。

【００６７】図６において、６は図４中の仮想分散ファイルサーバシステム４に相当する仮想分散ファイルサーバシステムである。この仮想分散ファイルサーバシステム６の特徴は、当該システム６を構成するファイルサーバ１０-1～１０-n内に、自サーバ（のストレージ装置１２-1～１２-n）に保持されている各ファイルについての負荷状況の情報（ファイル別負荷状況情報）を保持するファイル別負荷状況情報保持部１６-1～１６-nを備えて

いる点にある。これに伴い、仮想分散ファイルサーバシステム６の中心をなす仮想分散ファイルシステム６１の持つ機能も図４中の仮想分散ファイルシステム４１とは一部異なる。但し、仮想分散ファイルシステム６１内の各ファイルサーバ１０-1～１０-n毎の仮想分散ファイルモジュールには便宜的に図４と同一符号（４１０-1～４１０-n）を用いている。なお、図６では、モジュール４１０-1～４１０-n内の構成要素（仮想分散ファイルインタフェース、ローカルファイルインタフェース、通信モジュール）、及びファイルサーバ１０-1～１０-n上のマッピングテーブル、サーバ情報保持部は省略されている。

【００６８】ファイル別負荷状況情報保持部１６-i（i＝1～n）は、図７（a）に示すデータ構造を持ち、自サーバ内のファイル毎の負荷状況を示す情報と、そのファイルの属性を示す情報（ファイル属性）と、レプリケーションフラグとを含むファイル別負荷状況情報を保持する。ファイル属性は、対応するファイルがオリジナルであるかレプリカ（複製）であるかを示す。また、レプリケーションフラグは、対応するフラグがオリジナルの場合、そのレプリカを他サーバ側に生成済みであるか否か、つまりレプリケーション済みであるか否かを示す。

【００６９】図６には、ファイルサーバ１０-nが持つストレージ装置１２-n内に、ファイルサーバ１０-1が持つストレージ装置１２-1内の任意のファイル６１１のレプリカ６１２がレプリケーションＢ１により保持されている様子が示されている。

【００７０】次に、図６の構成の動作を説明する。仮想分散ファイルシステム６１上の各仮想分散ファイルモジュール４１０-1～４１０-nは、ファイル別負荷状況情報保持部１６-1～１６-nを例えば定期的に参照する。そしてモジュール４１０-1～４１０-nは、保持部１６-1～１６-nに保持されているファイル別の負荷状況情報から、自サーバ（のストレージ装置１２-1～１２-n）に保持されているファイルの中に、第１の閾値を超えた負荷のファイルが存在することを検出した場合、他サーバの１つに対して対応するファイルのレプリカを非同期に生成するレプリケーション動作を、例えばプライベート通信路５を介してのサーバ間通信により行う。ここでファイルの負荷状況は、当該ファイルへの要求の待ち行列（キュー）にある要求数、或いは当該ファイルの待ち状態にある要求の示すサイズの総和であり、要求を受け付ける毎と要求を処理し終える毎に更新される。また、レプリケーションの対象サーバには、図示せぬサーバ情報保持部に保持されているサーバ情報に基づいて、例えば負荷が最も低いサーバを選択すればよい。

【００７１】仮想分散ファイルモジュール４１０-1～４１０-nはレプリケーション動作を行うと、自サーバのファイル別負荷状況情報保持部１６-1～１６-nに保持されている対応するファイルの負荷状況情報中のレプリケーションフラグをレプリケーション済みの通知状態にセットする。またレプリケーション動作の対象となったサーバの仮想分散ファイルモジュールは、自サーバのファイル別負荷状況情報保持部内に対応するレプリカの負荷状況情報を追加する。

【００７２】ここでは、図６に示すように、ファイルサーバ１０-1の保持するファイル６１１のレプリケーションＢ１がプライベート通信路５を介してファイルサーバ１０-nに対して行われて、そのレプリカ６１２が当該ファイルサーバ１０-nのストレージ装置１２-nに保持されたものとする。この場合、ファイルサーバ１０-1のファイル別負荷状況情報保持部１６-1に保持されているファイル６１１の負荷状況情報中のレプリケーションフラグがレプリケーション済みを示す状態にセットされる。また、ファイルサーバ１０-nのファイル別負荷状況情報保持部１６-nには、ファイル６１１のレプリカ６１２についての新たな負荷状況情報が追加される。この負荷状況情報中のファイル属性は、対応するファイルが（ファイル６１１の）レプリカ（６１２）であることを示す。

【００７３】以後、クライアント２からファイル６１１の新たな読み出し要求があった場合、当該ファイル６１１を保持するファイルサーバ１０-1（の仮想分散ファイルモジュール４１０-1）は、当該ファイル６１１の負荷が第２の閾値（但し、第２の閾値＜第１の閾値）を超えているか否かを調べ、超えているならば、クライアント２からの要求に応答しない。この場合、クライアント２からの要求に対しては、レプリケーションを受けたファイルサーバ１０-nが応答する。ここでファイルサーバ１０-nは、ファイルサーバ１０-1が応答するか否かを考慮する必要はなく、要求されたファイル６１１のレプリカ６１２を有する限り、クライアント２に応答すればよい。

【００７４】このように、クライアント２からのファイル６１１に対する新たな読み出し要求を、そのレプリカ６１２を用いてファイルサーバ１０-nが処理することで、そのファイル６１１を保持するファイルサーバ１０-1では、それ以前に受け付けた当該ファイル６１１に対する読み出し要求の処理が進み、当該ファイル６１１の負荷が上記第２の閾値以下となる。するとファイルサーバ１０-1上の仮想分散ファイルモジュール４１０-1は、ファイルサーバ１０-n上の仮想分散ファイルモジュール４１０-nに対して、ファイル６１１のレプリカ６１２を消去するための要求を例えばプライベート通信路５を介したサーバ間通信により送る。

【００７５】この要求を受けたファイルサーバ１０-n上の仮想分散ファイルモジュール４１０-nは、既に受け付け済みの要求に対してのみレプリカ６１２を用いて処理を行い、しかる後にレプリカ６１２と対応する負荷状況情報を消去する。一方、ファイルサーバ１０-1上の仮想分散ファイルモジュール４１０-1は、クライアント２か

らのファイル６１１に対する新たな読み出し要求があれば、それに対して応答する。

【００７６】ところで、ファイルサーバ１０-1からファイルサーバ１０-nへのファイル６１１のレプリケーションにより、当該ファイル６１１に対する読み出し要求をファイルサーバ１０-nで受け付けるようになった結果、ファイルサーバ１０-1におけるファイル６１１の負荷が第２の閾値以下となる前に、ファイルサーバ１０-nにおける当該ファイル６１１のレプリカ６１２の負荷が第１の閾値を超えることがあり得る。

【００７７】そこで、このような場合、今度はファイルサーバ１０-nがレプリカ６１２を用いて次の世代のレプリカを他の１つのサーバに生成し、即ちレプリケーションのレプリケーションを行い、そのサーバでファイル６１１に対する読み出し要求を処理させればよい。そのためには、ファイル別負荷状況情報保持部１６-i（ｉ＝１～ｎ）に保持されるファイル毎の負荷状況情報に、図７（ａ）に示したような負荷状況とレプリケーションフラグに加えて、図７（ｂ）に示すように、ファイルの世代情報を持たせるとよい。

【００７８】この場合、ある世代のレプリカの負荷が上記第２の閾値以下に下がった時点で、当該レプリカを持つサーバから、そのサーバによるレプリケーションの対象となったサーバの持つ次世代のレプリカを消去する等の制御を行うことができる。このとき、レプリカの消去が要求されたサーバが、別のサーバに対して更に次世代のレプリカを生成している場合、その更に次世代のレプリカを消去するとよい。この他に、少なくともレプリカに関連したファイルの負荷状況情報については、前記したサーバ情報と同様に、各サーバ間の一致化を図ることにより、同一ファイル（レプリカを含む）について、負荷が最も低いファイルを持つサーバが、当該ファイルに対する読み出し要求に応答するようにしてもよい。

【００７９】最近では、ビデオ、オーディオ等のストリーミングデータや、或いはWWW（World Wide Web）のコンテンツ等、基本的には読み出しが主で、比較的サイズが大きく、ある程度のレスポンス（場合によっては帯域保証）が必要なデータが増加しつつある。しかもこうしたデータは、短期的に見て特定のデータ（ファイル）にアクセスが集中するケースが想定されるため、レスポンスを確保するのが困難な場合もある。以上に述べた図６の構成は、こうした状況を想定したもので、特定のファイルにアクセスが集中した場合に、自動的に当該ファイルのレプリケーションを行うことで、当該ファイルへのアクセスを分散させることができるようにしている。この構成は、単に負荷分散だけでなく、例えば重要性の高いファイルのバックアップに利用することも可能である。

【００８０】［第５の実施形態］図８は本発明の第５の実施形態に係る仮想分散ファイルサーバシステムを適用するコンピュータ・ネットワークシステムの構成を示すブロック図であり、図４と同一部分には同一符号を付してある。

【００８１】図８において、８は図４中の仮想分散ファイルサーバシステム４に相当する仮想分散ファイルサーバシステムである。この仮想分散ファイルサーバシステム８の特徴は、ファイルサーバ１０-1～１０-n、及びストレージ装置１２-1～１２-nが、例えばＦＣ－ＡＬ（Fibre Channnel Arbitrated Loop）８０により相互接続され、（ホストとしての）各ファイルサーバ１０-1～１０-nから（ターゲットとしての）ストレージ装置１２-1～１２-nの共有が可能な（つまりマルチホスト可能な）ネットワーク構成を適用している点にある。ここでは、図４の構成と異なって、プライベート通信路５を持たない点に注意されたい。

【００８２】この図８の構成では、図４の構成において（仮想分散ファイルモジュール４１０-1～４１０-nの通信モジュール４１３-1～４１３-nにより）プライベート通信路５を介して行われるサーバ間通信を、図１の構成と同様にネットワーク３を介して行えばよい（図は、この状態が示されている）。また、上記サーバ間通信を、ファイルサーバ１０-1～１０-nのストレージ接続用のインタフェースを介してＦＣ－ＡＬ８０上で行うようにしてもよい。この場合、プライベート通信路５を用いたのと同様に、ネットワーク３の負荷を軽減できる。

【００８３】図８の構成によれば、ストレージ装置１２-1～１２-nが全てのファイルサーバ１０-1～１０-nから直接に見えるので、各サーバ１０-1～１０-nに図６中のファイル別負荷状況情報保持部１６-1～１６-nを持たせることで、前記第４の実施形態で述べたようなレプリケーション動作や負荷分散を容易に行うことができる。なお、マルチホスト可能なネットワーク（インタフェース）はＦＣ－ＡＬ８０に限るものではなく、ＳＣＳＩ（Small Computer System Interface）バスであっても構わない。

【００８４】

【発明の効果】以上詳述したように本発明によれば、ネットワーク上に分散した複数のファイルサーバを、クライアントからは単一のサーバとして扱うことができ、サーバ台数やストレージ装置の接続状態をクライアントに意識させることがない。

【００８５】また本発明によれば、サーバを増設した場合、動的にボリュームを拡張することもできる。

【００８６】更に本発明によれば、複数のサーバ間で自律的な負荷分散が実現できる。

【図面の簡単な説明】

【図１】本発明の第１の実施形態に係る仮想分散ファイルサーバシステムを適用するコンピュータ・ネットワークシステムの構成を示すブロック図。

【図２】図１中のマッピングテーブルのデータ構造例を

19

示す図。

【図３】図１中のサーバ情報保持部のデータ構造例を示す図。

【図４】本発明の第２の実施形態に係る仮想分散ファイルサーバシステムを適用するコンピュータ・ネットワークシステムの構成を示すブロック図。

【図５】本発明の第３の実施形態に係る仮想分散ファイルサーバシステムを適用するコンピュータ・ネットワークシステムの構成を示すブロック図。

【図６】本発明の第４の実施形態に係る仮想分散ファイルサーバシステムを適用するコンピュータ・ネットワークシステムの構成を示すブロック図。

【図７】図６中のファイル別負荷状況情報保持部のデータ構造例を示す図。

【図８】同実施形態の動作を説明するタイミングチャート。

【符号の説明】

１，４，６，８…仮想分散ファイルサーバシステム

２…クライアント

20

３…ネットワーク

５…プライベート通信路

１０-1～１０-n…ファイルサーバ

１１，４１，６１…仮想分散ファイルシステム

１２-1～１２-n…ストレージ装置

１３-1～１３-n…ローカルファイルシステム

１４-1～１４-n…マッピングテーブル

１５-1～１５-n…サーバ情報保持部

１６-1～６-n…ファイル別負荷状況情報保持部

８０…ＦＣ－ＡＬ（マルチホスト可能なインタフェース）

１１０-1～１１０-n，４１０-1～４１０-n…仮想分散ファイルモジュール（管理モジュール）

１１１-1～１１１-n…仮想分散ファイルインタフェース

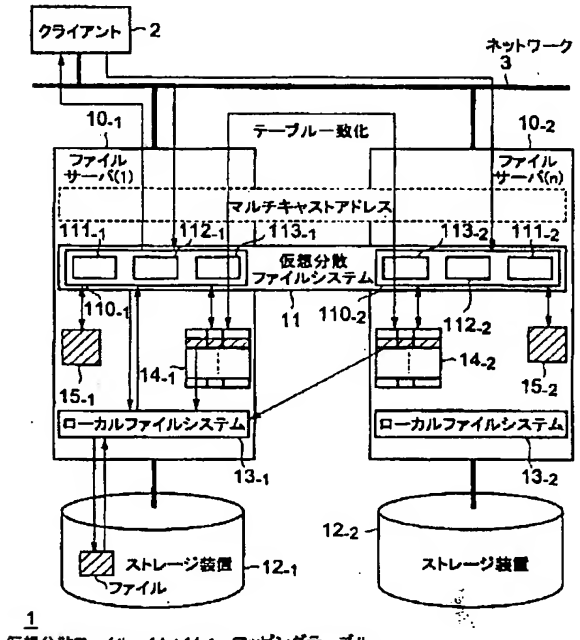
１１２-1～１１２-n…ローカルファイルインタフェース

１１３-1～１１３-n…通信モジュール

６１１…ファイル

６１２…（ファイル６１１の）レプリカ

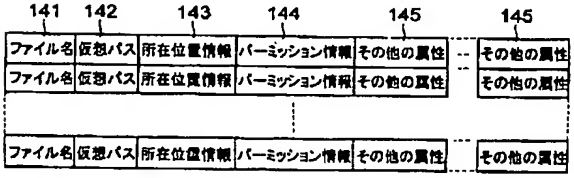
【図１】



1
仮想分散ファイル
サーバシステム

14-1,14-2…マッピングテーブル
15-1,15-2…サーバ情報保持部
110-1,110-2…仮想分散ファイルモジュール
111-1,111-2…仮想分散ファイルインタフェース
112-1,112-2…ローカルファイルインタフェース
113-1,113-2…通信モジュール

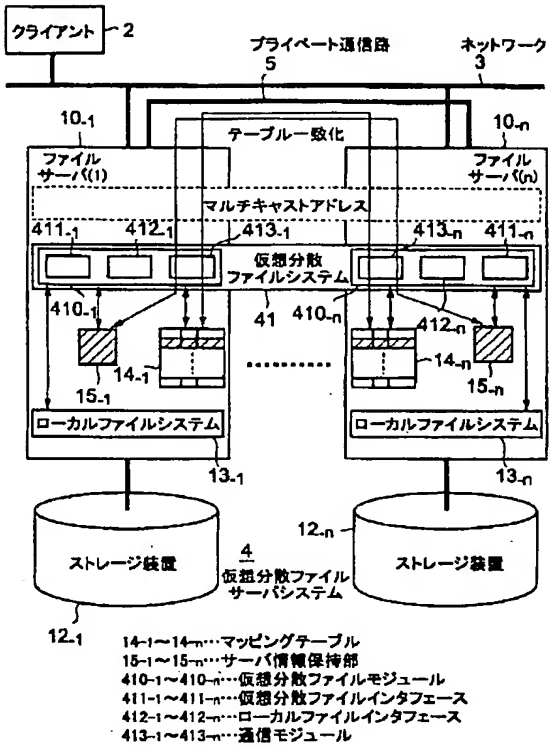
【図２】



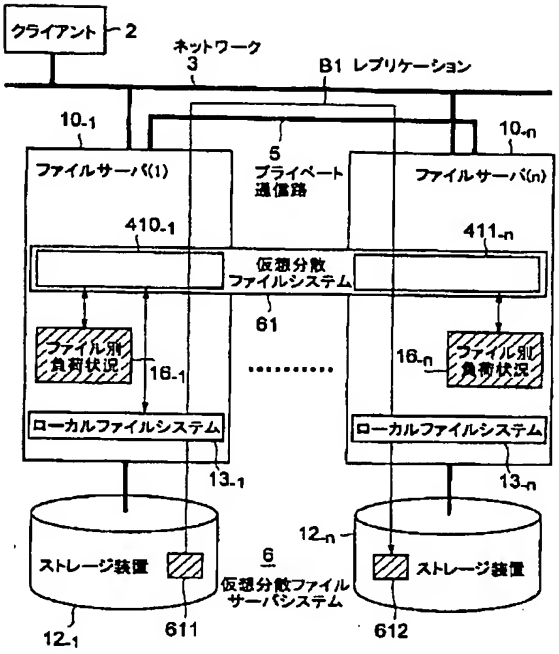
【図３】

サーバID	リソース情報	負荷状況
サーバ1		
サーバ2		

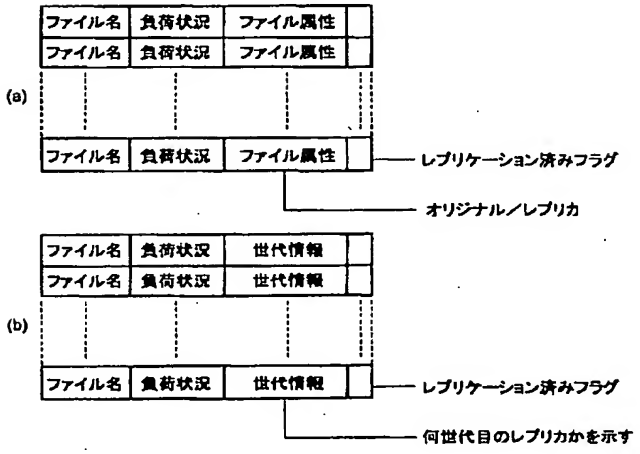
【図４】


クライアント 2
プライベート通信路 5
ネットワーク 3
10-1
10-n
テーブル一致化
ファイルサーバ(1)
ファイルサーバ(n)
マルチキャストアドレス
411-1
412-1
413-1
413-n
411-n
仮想分散ファイルシステム
41
410-1
410-n
412-n
14-1
14-n
15-1
15-n
ローカルファイルシステム
13-1
13-n
12-n
ストレージ装置
4
仮想分散ファイルサーバシステム
12-1
14-1～14-n…マッピングテーブル
15-1～15-n…サーバ情報保持部
410-1～410-n…仮想分散ファイルモジュール
411-1～411-n…仮想分散ファイルインタフェース
412-1～412-n…ローカルファイルインタフェース
413-1～413-n…通信モジュール

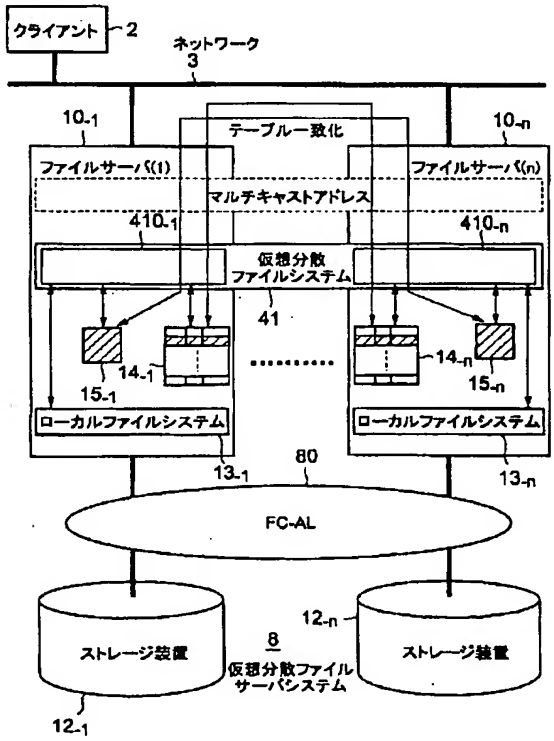
【図６】


クライアント 2
ネットワーク 3
B1 レプリケーション
10-1
10-n
5
プライベート通信路
ファイルサーバ(1)
ファイルサーバ(n)
410-1
411-n
仮想分散ファイルシステム
61
ファイル別負荷状況
16-1
16-n
ローカルファイルシステム
13-1
13-n
12-n
ストレージ装置
6
仮想分散ファイルサーバシステム
12-1
611
612

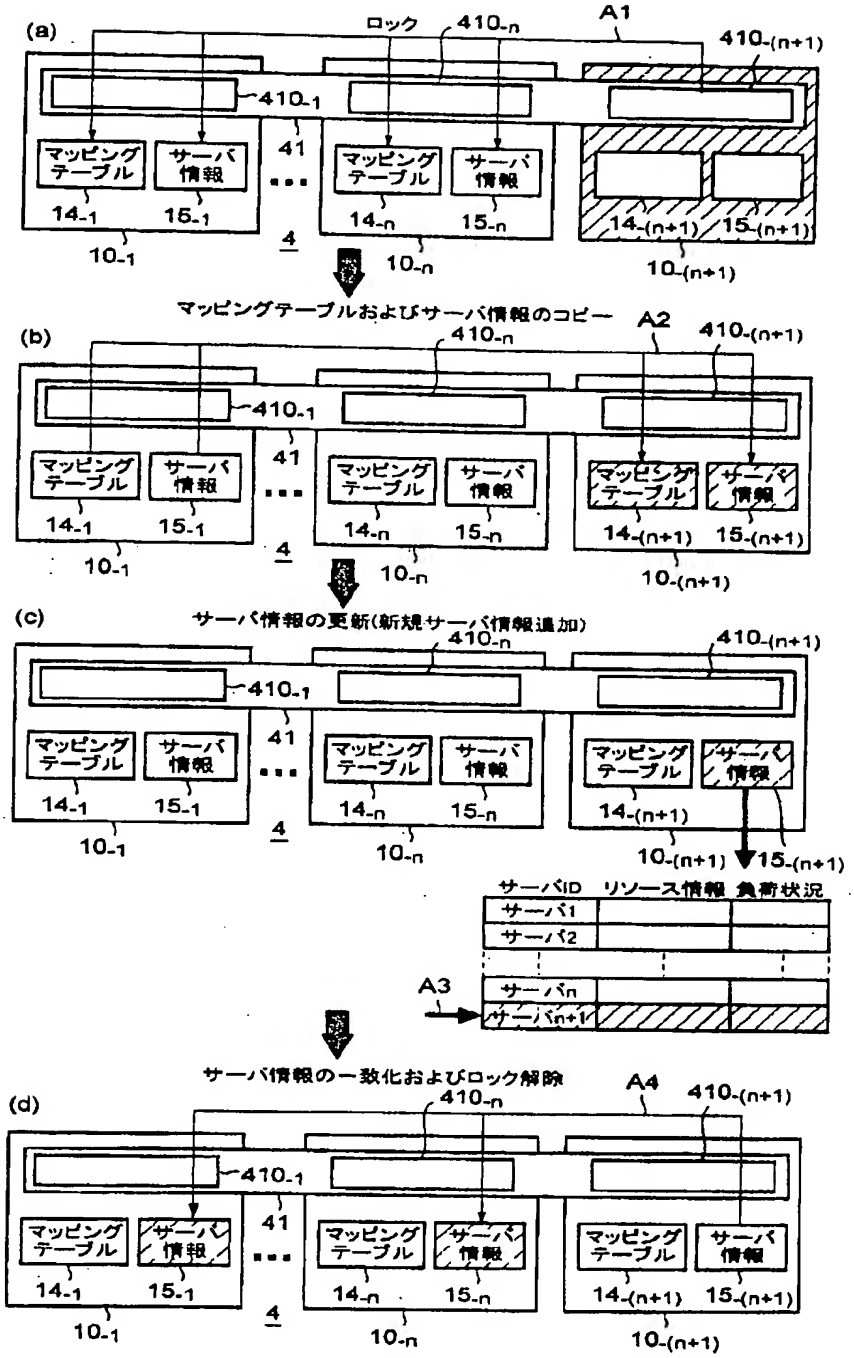
【図７】


ファイル名
負荷状況
ファイル属性
(a)
レプリケーション済みフラグ
オリジナル／レプリカ
世代情報
(b)
何世代目のレプリカかを示す

【図８】


クライアント 2
ネットワーク 3
10-1
10-n
テーブル一致化
ファイルサーバ(1)
ファイルサーバ(n)
マルチキャストアドレス
410-1
410-n
仮想分散ファイルシステム
41
14-1
14-n
15-1
15-n
ローカルファイルシステム
13-1
80
13-n
FC-AL
12-n
ストレージ装置
8
仮想分散ファイルサーバシステム
12-1

【図５】


(a)
ロック
410-n
A1
410-(n+1)
410-1
マッピングテーブル
サーバ情報
41
14-1
15-1
14-n
15-n
14-(n+1)
15-(n+1)
10-1
4
10-n
10-(n+1)
マッピングテーブルおよびサーバ情報のコピー
(b)
A2
(c)
サーバ情報の更新(新規サーバ情報追加)
A3
サーバID
リソース情報
負荷状況
サーバ1
サーバ2
サーバn
サーバn+1
サーバ情報の一致化およびロック解除
(d)
A4

フロントページの続き

Ｆターム(参考) 5B082 CA18 EA07 HA03 HA05 HA08
HA09
5B089 GA12 JA11 JB15 KA00 KC15
KC28 KE07